By contrast, many functions within the host digital processors 12 inherently utilize physical device names or addresses to identify attached storage devices. For example, the plug-and-play manager within a Windows™ 2000 host identifies storage devices via physical device object names that include, among other things, port number, path number, target number and logical unit number.

The illustrated embodiment provides a mechanism for readily associating these physical device names/addresses with the corresponding LUN IDs, thereby, facilitating use of built-in host functions – e.g., plug-and-play manager detection services – to determine when the SAN storage devices have been added, removed, enabled, disabled, otherwise affected. Though the discussion here focuses on association of physical device object names of the type used by plug-and-play managers in the Windows™ 2000 environments, those skilled in the art will appreciate that the teachings are equally applicable to forming other such associations with this and other operating systems and operating system functions.

Referring to FIGURE 36 by way of review, in normal operation of a Windows™ 2000 host, the plug-and-play manager (PNP) queries the SCSI port driver 356 for information regarding all devices known by it. The information includes data such as port number, path number, target number, and logical unit number for each found device. The PNP manager 386 generates from this a physical object for each device.

Subsequently, when the PNP manager 386 detects that a storage device has been added or removed, e.g., coupled or decoupled from the interconnect 16, it generates an event. In a Windows™ 2000 environment, this is referred to as a "device change" event and includes a physical device object name, to wit, a string with the host bus adapter (HBA) name, port number, path number, target number, and logical unit number of the affected device. In embodiments operating on hosts with other operating systems, such an event may have a different name and/or content.

A user mode process executing on the host receives such PNP events, so long as that process is appropriately registered with the PNP manager. The process extracts the port number, path number target number, and logical unit number from the physical device object name and converts them to a form suitable for querying the device or its interface (e.g., the port driver and/or HBA) adapter for SCSI inquiry data, e.g., of the type contained on Page 83h and/or Standard Page. It uses this to open a handle to the device and obtain that SCSI inquiry data, e.g., by way of an IOCTL_SCSI_GET_INQUIRY_DATA call in the Windows™ 2000 environment or using a related or analogous call in other environments.

Using the SCSI inquiry data and the information extracted from the physical device object name, the user mode process generates an LUN ID using the algorithms discussed above in connection with FIGURE 10. In this manner, it thereby forms an association between a physical device object name and logical identifier, to wit, a LUN ID.

In the illustrated embodiment, the user mode process forms such an association, e.g., for purposes of correlating the LUN ID included in a storage device assignment received from the SAN manager 20 with events generated by the host PNP manager. In this regard, the user mode process executes the algorithm identified within the LUN ID of the assigned device in order to convert the inquiry data and extracted information into a logical identifier. In alternate embodiments, the user mode process can exercise this or other LUN generation algorithms, e.g., for purposes of matching a raft of identified LUN IDs or for other purposes. In the illustrated embodiment, the aforementioned user mode process is a PNP event listener, though it can comprise any code operating in user mode. Moreover, the mechanism discussed above can be used to associate a physical device name or address of any device (disk or otherwise) with a logical identifier.

Fiber Channel Device Determination in Kernel Mode

As discussed above, in order to mask non-assigned LUNs, the filter driver 354 intercepts claim requests made by the class driver 352 to the port driver 356 or, conversely, the port driver response to those claims. For such claims, the filter driver compares the identified devices against the LUN IDs listed in the data table 354a, applying the associated LUN generation algorithms and comparing the results to determine whether the response should be passed or blocked. Because the filter driver 354 executes in kernel mode in Windows™ NT, Windows™ 2000 or other such hosts, operating system, adapter or storage device limitations may preclude the driver 354 from consistently determining whether any given claim is for a fiber channel device and, hence, subject to potential masking.